Phase 4 Report

The Game

Description:

In a land and time far, far away from our own, a demon smiled wickedly as he looked down at the blueprints in his hands. No one knew of the great demon that would become the king of the world, but they soon would! . . . Just as soon as he infiltrated the nasty heroes' base!

Play as the debutant demon and steal all the heroes' treasures to make your mark in history. Watch out for any demon-catching traps in the heroes' base, and make sure you don't get caught by the heroes themselves! Try to grab their elusive, most-prized treasure for even more fame and fortune.

Plan and design:

Although we remained as loyal as we could to our game's original design and plan, some modifications were necessary to fulfill our goals. We made certain changes to specific classes, such as the Coordination class, which we integrated into the gameObj class to include additional methods that its reliant classes all needed. Additionally, we established a dependency between the gameObj class and map objects. Furthermore, we eliminated the Square class and developed a bitmap for our game map, which was needed in our Pathfinder class to make our moving enemies follow the player character and increase our game's entertainment. These alterations facilitated code comprehension and class connectivity, while also enabling easier scalability of the project in the future.

Since this project was the first time that any of our team members had made a game like this, the final product of our work ended up including methods we did not have in our original UML diagram. For example, draw() functions in each of the classes that needed to be displayed on our game screen. Another example would be the implementation of threads in some of our classes, like the GameManager. We learned as we worked.

One of the most significant changes that needed to be made, however, was a redistribution of tasks and responsibilities due to the loss of a team member. As the team member had a critical role in the project, the remaining team members needed to pick up the slack and take on additional tasks to ensure the project's successful completion. This required a reevaluation of the project timeline and our milestones, with adjustments made to accommodate the increased workload.

Another change that occurred was a shift in the project's focus. As the team member who dropped out was responsible for a particular aspect of the project, such as overall design of classes, the team needed to adjust their approach to account for the missing skill set. This resulted in changes to the project's design, with a shift in priorities and a greater emphasis placed on different aspects of the software. We decided against implementing the pause game

feature and multiple levels, as that required time that we no longer had. In addition, we did not have as much time to spend on the visuals of our game as we had hoped, so we had to depend on online resources for our sprites instead of creating our own. The limitations of the royalty-free, free-to-use images online meant that we needed to adjust our game description to fit our game's final look.

Overall, losing a team member in a software project was challenging, but with careful planning and communication, the team was able to adapt and continue to make progress towards our goals. It was essential to remain flexible and adaptable, focusing on the project's priorities and working collaboratively to overcome any obstacles that arose.

Most important lessons learned:

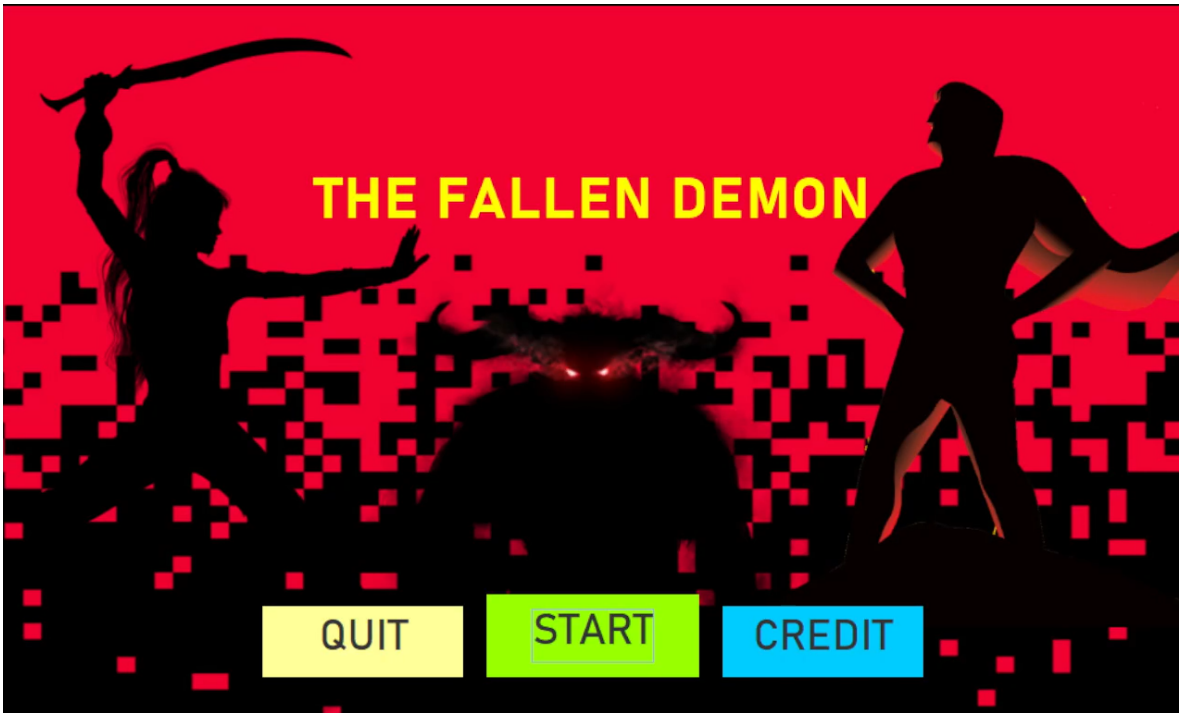
Working on this project gave us valuable learning experiences and we learned important lessons from it. Here are the most important lessons we learned:

- Collaboration: Working in a group of three required collaboration and communication to ensure that everyone was on the same page. It was essential to delegate tasks, cooperate towards meeting deadlines, and work together to complete the project efficiently.
- Project organization: Using Maven helped to manage project dependencies and simplify the build process. Learning to use Maven effectively assisted making the project more manageable and easier to maintain.
- Object-oriented programming: Working on a project with multiple classes and objects helped us to reinforce the principles of object-oriented programming. It was essential to ensure that each class was well-designed and had a clear purpose.
- Testing: Writing unit and integration tests helped to ensure that the code was working as expected and could catch bugs before they became significant issues. The code review that we performed as a part of our testing phase also helped to ensure our code's quality.
- Documentation: Writing clear and concise documentation could help to ensure that the code was easy to understand and maintain. It was also essential to document how the application could be compiled, built, and run to provide ease of usage.

Overall, working on a maze game project using Java with Maven in a group of three could provide valuable learning experiences that could be applied to future projects.

Tutorial:

Upon opening the game, a menu screen will pop-up like the one below:

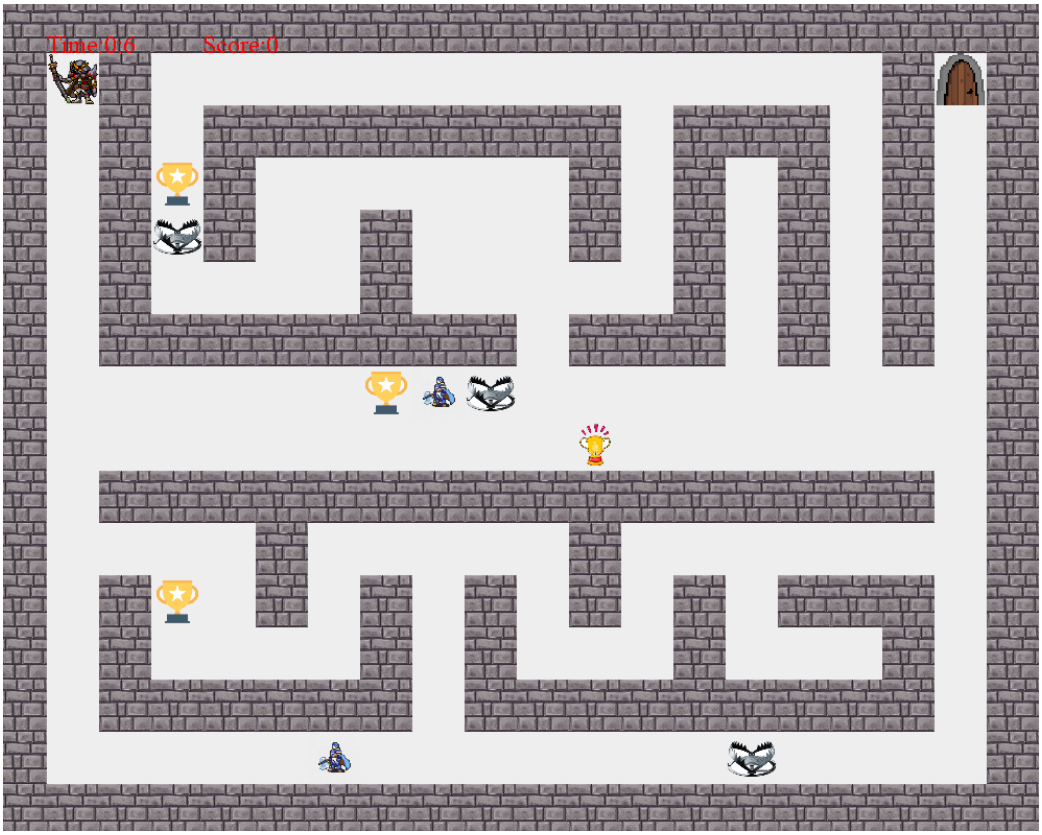


QUIT ends the application and closes the window.
START brings you to the game.
CREDIT shows you the names of the people who worked on the game.

In the game, you play as a demon hoping to become famous, and who looks like this:

Use the WASD keys on the keyboard to move him up, left, down, and right, respectively.

When you press START to play the game, you will see something like this:



You can see your score and the amount of time you've been playing 1 round of this game in the top left corner of the screen.

The goal of the game is to get to the door:

You will need to travel through the maze to get to it, the walls of which look like this (you cannot walk on them):

However, the door is locked. You need to collect all the regular rewards (the treasure of the heroes), which look like this:

in order to open the door and successfully escape the heroes' labyrinthine base (win the game!). If you do not have all the regular rewards, the door will not open. After all, what would be the point of infiltrating the heroes' base and going through all this trouble? (Go big or go home!) Each regular reward will give you 100 points of fame.

This is a bonus reward (the heroes' most-valuable prize):

For an extra 200 points, grab the bonus reward as well. But be quick about it! The elusive bonus reward will disappear after a few seconds. You do not need to collect this to open the door.

Be careful as you traverse the maze and collect the rewards, though, because there are two types of enemies in your way: traps and heroes.

The stationary traps look like this:

Avoid them when you can, because they will decrease your score by 110 points (the demon has to leave some treasure behind to escape from a trap!), and if your score drops below 0, the game will end and you will lose.

The heroes (moving enemies) look like this:

They will constantly move towards the player (thanks to ~~our path-finding algorithm~~ their terrifyingly-good, demon-catching senses)! If they catch you, the game ends immediately (as well as the demon's career).

Please refer to the gif on our repository's README for gameplay examples. There are demonstrations of winning the game (and showing how you cannot open the door without collecting the regular rewards), of losing the game due to stepping on too many traps, and of losing the game after being caught by the moving enemies.